International Journal of Research in Computer Science
eISSN 2249-8265 Volume 2 Issue 3 (2012) pp. 7-10
© White Globe Publications
www.ijorcs.org

A NOVEL APPROACH TO GENETIC ALGORITHM BASED CRYPTOGRAPHY

Farhat Ullah Khan[1], Surbhi Bhatia[2]
[1]*Assiatant Professor, Amity University*
Email: fukhan@amity.edu
[2]*Student M. Tech, Amity University*
Email: surbhibhatia1988@yahoo.com

Abstract: *Cryptography is immensely essential ingredient of network security. Public Key Cryptography, one of the most important forms of cryptography, requires the key to be unique and non-repeating. There are two ways of producing the key. One is rigorous mathematically strong algorithmic approach like AES and the other is the approach that mimics nature. The work presented explores various attempts that have been made in this direction and suggests a new technique using Genetic Algorithms. The technique has been implemented and analyzed. The results obtained are encouraging. The samples satisfy most of the tests including gap test, frequency test etc thus strengthening the belief that the algorithm is as strong, if not better than any of the mathematically strong approach.*

Keywords: *Cryptography, Genetic Algorithms, One Time Pad, Vernam Cipher.*

I. INTRODUCTION

Genetic Algorithms (GAs) being optimization algorithms unite ‘survival of the fittest’ and a simplified version of Genetic course [1]. A thorough study on the success of GAs in cryptography was carried out by Benthany Delman [2]. The work takes GA as base for generating the key and proposes a novel technique to produce a key which can substitute One Time Pad (OTP) in the Vernam Cipher. The key produced is non-repeating and thus making the cipher almost unbreakable. GAs in spite of being random have the ability to make the population converge to the desired point using a fitness function [1].The technique has been implemented and the randomness of the population generated was calculated. The experiments carried out recognized the ability of GAs to produce a good quality random sample. If the key of the Vernam Cipher is selected from that sample then it is found to be better as compared to existing PRNG. Many attempts have been made to accomplish the above task using GAs. Some of them open the door for further exploration in the field. The work carries forward one such attempt [1] and analyses the effect of the changes proposed. The initial population taken in the work is binary. Standard Genetic operators like mutation and crossover have been applied to the population to improve the quality of the sample. The technique proposed has been explained in the following section and results have been explicated. The work produces a decent sample satisfying most of the test. In the future we intend to check the strength of key generated by exposing the cipher text to various attacks.

II. CRYPTOGRAPHY

The application of Genetic Algorithm and cryptography has been discussed in some of the works [1], [2]. In most of the cases GA approach has been applied to decrypt simple ciphers. In the security analysis cipher attacked included monoalphabetic substitution cipher, transposition, permutation, vernam cipher etc.

Monoalphabetic Substitution Cipher: A key consist of all the possible permutation of an alphabet which when replaced may lead to the deciphering of the text. To handle the problems of monoalphabetic substitution, other methods like polyalphabetic substitution and permutation, transposition cipher are used. The permutation cipher is applied to a block of ciphers while columns transposition is applied to the entire text at once.

Vernam Cipher- It is a stream cipher in which a binary key string of the same length to produce a cipher text such that

$$c[i] = p[i] + k[i]$$

Where c[i] = i^{th} character of the cipher text, k[i] = i^{th} character of the key, P[i] = i^{th} character of the plaintext.

In the work proposed, the intention is to create a key as strong if not stronger than the vernam cipher. If the key is randomly chosen and never used again, the cipher is called one time pad [1], [2]. The one time pad is theoretically unbreakable [1], [2], [3]. The cryptanalysis can at maximum guess the key in which case a very large number of guesses for the key to be correct can ensure the near unbreakability of the cipher [4].

III. SECURITY OF THE KEY

In the literature review, it was found that the characteristics that determine the strength of the key are not quantifiable but matrices might be used for evaluating and comparing cryptographic algorithm [5]. The characteristics that are considered are

Type: Symmetric or Asymmetric; Functions: Integrity and authentication of message; Key size and rounds; and the complexity of the algorithm. The attacks that can be carried out so as to test the strength of the algorithm are brute force, factoring and differential cryptanalysis. The matrices that have been used to judge the effect of these attacks are based on the key length and complexity of the algorithm.

IV. GENETIC ALGORITHMS

Genetic Algorithms are adaptive search procedures which are footed on Charles Darwin theory of the survival of the fittest [6]. The concept behind these algorithms was to imitate the randomness of the nature. So, GAs follows nature to a great extent. GAs produce a population in such a way that the attribute which is trendy, that is, has higher fitness value is replicated more, as is done by the nature. This is also the fundamental concept behind evolution. So, these algorithms are also referred as the evolutionary algorithms.

Genetic Algorithms (GAs) are search procedures to converge to optimal solution based on the theory of survival of the fittest. The basic entity of GA is chromosome. Each chromosome symbolizes a solution to the problem and is composed of a string of cells of finite length. The binary alphabet {0, 1} is often used to represent these cells but integers can be used depending on the application. The fitness value is a function or rationale against which chromosome is tested for its suitability to the problem in hand.

A. *Steps In Genetic Algorithms*

A brief overview of the steps involved in Genetic Algorithms is as follows.

Step 1. A population having P individuals are randomly generated by pseudo random generators whose individuals may represent a feasible solution. This is a representation of solution vector in a solution space and is called initial solution. This guarantees the search to be unbiased, as it starts from wide range of points in the solution space.

Step 2. Individual members of the population are evaluated to find the objective function value.

Step 3. In the third step, the objective function is mapped into a fitness function that computes a fitness value for each member of the population. This is followed by the application of GA operators.

1) Reproduction Operator: Reproduction is done on the basis of Rowlett Wheel selection .It selects chromosomes from the initial population and enters them into the mating procedure.

2) Crossover Operator: Crossover Rate (0 to 1) determines the probability of producing a new chromosome form the parents. For example, the strings 10000100 to 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 to 11100100. The crossover operator roughly mimics biological recombination between two single-chromosomes (haploid) organisms.

3) Mutation Operator: It randomly changes its genetic makeup. This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g., 0.001) [6], [7], [8].

V. PROPOSED WORK

The process of generating the key from the Genetic Population has the following steps

In the first step a binary population is generated. Each cell is generated using the pseudo random number generator of the programming language. The number generated is one if the PRNG generates a number greater than 50 else it is 0. Each chromosome contains 25 such cells and the number of chromosomes in the experiment was taken as 1000.

Now for each chromosome we repeat the following process:

Divide the chromosome into 5 groups.

Calculate the number of ones in each group.

If the number of one’s is greater than 2 then the new array will have 1 as its cell otherwise 0.

The above step converts the population of chromosomes having 25 cells as one having 5 cells. Now we have a 5X 1000 array with us.

The array is then read vertically 25 cells at the time. The first column followed by the second and so on.

The above step gives us an array of 25 X 200 which serves as the population now. This is followed by crossover and mutation operators being applied to the sample.

Now each cell is multiplied by $2^{(12-i)}$ where ’I’ is the cell number. This generates a sample of 200 numbers.

Each number is then converted into an integer. The process is repeated 5 times. The coefficient of auto correlation is then calculated. If the result is favorable then the population is accepted else the whole process is repeated.

VI. EXAMPLE

If the original population was a 5*5 population where the cells are generated randomly.

10010
10101
11101
00101
01011

The above population is read vertically. New array would be

11100
00101
01110
10001
01111

Any two random numbers are generated from 0 to 5, say, 2nd and 4th chromosome.

One point crossover is performed as shown below:

00101
10001

Taking the crossover point, COP= 2. The chromosome becomes

00001

After crossover has been performed, then mutation is done by selecting a random chromosome and flipping from amongst those chromosomes.

11110

The number of times the crossover is to be performed is given by the formula: Number of crossovers = Number of cells in each chromosome* Number of chromosomes* Crossover Rate/ 100

The number of times the mutation is to be performed is given by the formula: Number of Mutations = Number of cells in each chromosome* Number of chromosomes* Mutation Rate/ 100

Once the task is accomplished, then Coefficient of Correlation of the above is calculated by taking k=1, 2, and 3.

The coefficient of autocorrelation is defined as follows.

Given measurements, Y_1, Y_2, ..., Y_N at time X_1, X_2, ..., X_N, the lag k autocorrelation function is defined as

$$r_k = \frac{\sum_{i=1}^{N-k}(Y_i-\bar{Y})(Y_{i+k}-\bar{Y})}{\sum_{i=1}^{N}(Y_i-\bar{Y})^2} \qquad [11]$$

If the Coefficient of Correlation is satisfactory, then random chromosome is selected which is taken as key otherwise, the process is repeated.

VII. RESULTS AND CONCLUSIONS

The work has been implemented and analyzed. The implementation has been done in C#. Samples have been collected and analyzed in Microsoft Excel, Various tests have been applied on the sample and most of them give satisfactory results.

Since, around a 400 values were analyzed and no repetition was obtained therefore frequency test was not applied. The coefficient of autocorrelation was calculated for k = 1 to k = 10. The result for k = 1 was 0.03, thus indicating a good random sample. Karl Pearson Coefficient of correlation has been calculated, also giving satisfactory data.

In the analysis of data, a correlogram is an image of correlation statistics. In time series analysis, a correlogram, also known as an autocorrelation plot, is a plot of the sample autocorrelations r_h versus h (the time lags). The correlogram is a commonly used tool for checking randomness in a data set. This randomness is ascertained by computing autocorrelations for data values at varying time lags. If random, such autocorrelations should be near zero for any and all time-lag separations. If non-random, then one or more of the autocorrelations will be significantly non-zero.

The majority was calculated by taking a group of 10 cells. If more samples are needed then a set of 5 cells can also be taken. The whole process needs to be enhanced, analyzed and put to more stringent tests. Moreover it was also observed that the sample obtained distribute almost equally between data samples.

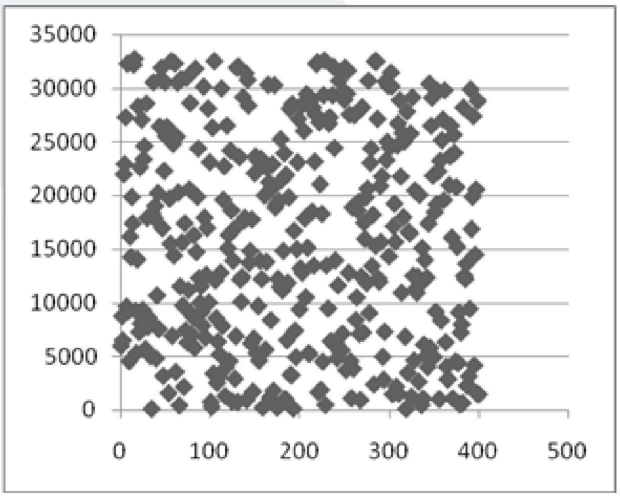

Figure 1: Scatter Diagram of Samples Obtained

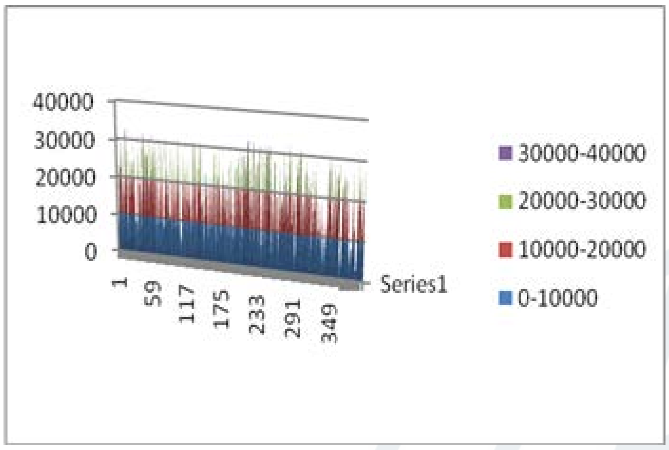

Figure 2: Sample division

VIII. FUTURE SCOPE

The task has been accomplished and tested. The tests are done using the Coefficient of autocorrelation as the principle factor in determining the randomness of the sample. Since, it is a process based on heuristic selection; its strength can be compared with the contemporary mechanism of producing keys based on Cellular Automata, Corpuscular theory and ACO. Random Number Generation via Cellular Automata [9] has been proposed in some paper. The task therefore is to implement those works and compare the above work with a Cellular Random Number Generator. Corpuscular theory [10] is a relatively new one. Its results are being asked for and when available will be compared with the above work. PRNG is using ACO will be developed in the next phase.

IX. REFERENCES

[1] Harsh Bhasin, Nakul Arora, Reliability Infocom Technology and Optimization 2010, Conference Proceedings pages 226- 230.

[2] Bethany Delman, Genetic Algorithms in Cryptography, MS Thesis 2004.

[3] ABDELSALAM ALMARIMI et al, A NEW APPROACH FOR DATA ENCRYPTION USING GENETIC ALGORITHMS, Published in: · Proceeding CERMA '10 Proceedings of the 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference

[4] Menezes, A., van Oorschot, P., & Vanstone, S. (1997). Handbook of Applied Cryptography Boca Raton: CRC Press

[5] Norman D. Jorstad, CRYPTOGRAPHIC ALGORITHM METRICS, January 1997

[6] Harsh Bhasin, Surbhi Bhatia, Use of Genetic Algorithms for Finding Roots of Algebraic Equations, IJCSIT, Volume 2, Issue 4, Pages 1693-1696

[7] Harsh Bhasin, Supreet Singh, GA-Correlation Based Rule Generation for Expert Systems, IJCSIT, Volume 3, Issue 2, Pages 3733-3736

[8] Harsh Bhasin, Surbhi Bhatia, Application of Genetic Algorithms in Machine learning, IJCSIT, Volume 2, Issue 5, Pages 2412-2415

[9] Stephen Wolfram, Cellular Automata And Complexity: Collected Papers, 1994, ISBN 0-201-62716-7.

[10] Thomson, J. J. (Joseph John), 7 Mar 2010, The corpuscular theory of matter .

[11] Raymond Kan, Xiaulo Wang (November 2008), On the Distribution of the sample Autocorrelation Coefficients.